



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Expert System Linked to a Simulation Model in the Expert Of Decision Making

B. Venkata Raju

Lecturer, Department of Computer Applications, Jeppiaar Engineering College,
Old Mamallapuram Road, Chennai-600 119, India

Abstract: In this research an expert systems approach to representing the human decision maker is proposed. An example of an expert system linked to a simulation model is given. Not only was it possible to train the expert system by programming a set of decision rules, but also by obtaining examples of decisions through running and interacting with the simulation model.

Key words: Simulation models, human decision making, expert system

INTRODUCTION

Many simulation models include some elements of human decision making. Typically these involve scheduling and allocation rules. For instance, a production supervisor needs to determine a week's production schedule, or to consider which staff to allocate to particular tasks at the start of a shift. In rail yard operations, goods wagons are allocated to particular sidings by the yard supervisor. In warehouse and distribution operations the allocation of stock is often determined by a supervisor, as well as the allocation of lorries to loading/unloading bays. Customers determine which route to take around a supermarket based on a complex set of decision rules. Indeed, since most simulation models represent human activity systems, as opposed to purely automated systems, there is almost always some element of human decision making.

In this study we discuss how expert systems could be used to represent these elements of human decision making in simulation models.

Expert systems in the life-cycle of simulation studies: It has been proposed that expert systems could aid the development and use of simulations throughout the life-cycle of a simulation study (Doukidis and Angelides, 1994). Indeed, there are examples of expert systems being applied at every stage of a simulation study, from model conception to experimentation and the analysis of results. An early attempt at automating the development of conceptual models can be found in Doukidis and Paul (1985). Later, however, it is conceded that intelligent front ends probably provide a less rigid and, therefore, more useful approach (Doukidis and Angelides, 1994).

Input data modeling provides a role for expert systems in the simulation life-cycle. Hurrion (1993a) proposes that the approach might be used to generate random variants for a simulation model. In terms of model development, there have been attempts at using expert systems to automatically generate simulation program code, for instance, CASM (Balmer and Paul, 1986) and Mathewson (1989). Expert systems have also been used for model verification and validation. Doukidis (1987) uses an expert system, SIPDES, to help locate and resolve compilation errors in simulation programs. Deslanders and Pierreval (1991) develop a system with limited capability for aiding model validation.

As an aid to experimentation and results analysis, there is considerable scope for applying expert systems. For instance, Hurrion (1991) uses an expert system to aid the design of experiments. He also employs a neural network to analyze a simulation model's output.

Representing human decision making: The presence of human decision making within simulation models presents two problems to the simulation modeler. First, it is necessary to determine the way in which the decisions are made by the people involved and second, it is important that the decision making process is modeled as accurately as possible. It is probably the first of these that presents the greatest problem. For instance, when one of the authors (Robinson) was investigating the modeling of an engine assembly facility, it became apparent that it was all but impossible to determine how different supervisors allocated staff to machines on different shifts. What was apparent though was that some supervisors were more effective than others in their allocation decisions.

The typical approach to representing human decision making in simulation models is to try to elicit the decision rules from the decision maker. In some cases this amounts to little more than a guess on the behalf of the modeler. Following this, the rules are included in the model using the constructs of the simulation language or simulator. This normally requires the use of a series of if, then, else statements. This can result in large amounts of code that is difficult to interpret and even harder to change.

One approach to overcoming these problems might be to use an expert system to represent the human decision maker and link it with a simulation model. Indeed, some have already attempted to do this (Flitman and Hurriion, 1987; O'Keefe, 1989; Williams, 1996; Lyu and Gunasekaran, 1997). This could be implemented in two ways:

- Elicit the decision rules from the expert and represent them within an expert system.
- Use the simulation model to prompt the expert to make decisions, building up a set of examples from which an expert system could learn.

These correspond to the two fundamental approaches to knowledge acquisition for any expert system: elicitation by knowledge engineer and machine learning from examples, respectively. The first approach would employ the constructs of an expert system and so make it easier to accurately represent the decision process. It should also be easier to interpret and easier to change since expert systems are specifically designed to facilitate this. In this way the approach should aid model development. What it does not provide, however, is a simple means for knowledge elicitation. This remains a well-known problem in expert systems generally (Waterman, 1986). It is in the second approach where the link to a simulation model could provide significant advantages. Most work on machine induction (Hart, 1987) treats the set of examples as somehow given and devotes little or no discussion to the process of obtaining the examples. By getting the simulation model to present the human decision maker with realistic conditions and asking for a decision, a set of examples could be obtained at an accelerated speed (assuming the model runs faster than real-time!). In this way the approach acts as an aid to obtaining input data, that is, the process by which a human decision maker works.

With recent advances in computing technology, particularly Object Linking and Embedding (OLE), it should be relatively simple to run a simulation model and an expert system in parallel on the same PC. The next section describes an example of this approach, in which

an expert system is first programmed to represent the human decision maker and is then trained via examples obtained from the simulation.

Example application

Allocating buses at a boarding bay: Using the Witness simulation package a model was developed of a fictional bus stand as shown in the following Fig. 1.

Ten minutes (based on a negative exponential distribution) and require 10 to 40 Passengers to board (uniformly distributed). On arrival, the Buses are allocated to a suitable bay by the bay supervisor. In making this decision the supervisor must take account of the restrictions on the bay capacities. Buses requiring more than 20 Passengers must be allocated to bay 2 or 3, since bays 1 and 4 only have capacity for up to 20 Passengers. Should a bay not be available then the Bus waits in the park until a suitable bay becomes available. Once a Bus is allocated, it moves to the bay where the Passengers will be boarding before departing from the system. Buses take one minute to move to the boarding bay where each Passenger takes one minute to be board.

Linking an expert system to the simulation model:

XpertRule was used to develop the expert system that represents the supervisor's allocation decisions. This package was selected for two reasons. First, it adopts a rule induction approach. Consideration was given to both neural network and case based reasoning approaches, but while these may provide benefits in terms of their ability to learn from examples, neither is able to provide information on how decisions are taken. Since, as discussed in the conclusion, this could be an important benefit of using expert systems, a rule induction approach was considered most appropriate. Second, XpertRule is one of the few expert systems packages available that has a true Windows implementation and is OLE compliant.

Since Witness can only work as an OLE slave, it was necessary to develop a Model Controller (MC) in Visual Basic as shown below Fig. 2. The MC initiates the run of the simulation model. At a point where an allocation decision is required, the simulation model automatically

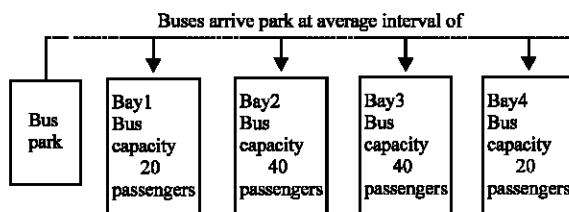


Fig. 1: Bus boarding Bay example

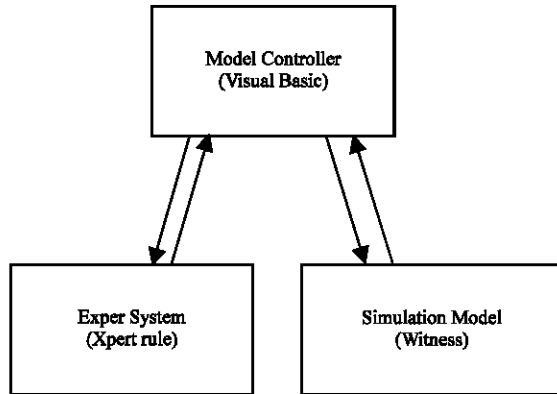


Fig. 2: Linking witness to XpertRule

stops and waits until the MC returns a decision and continues the run. Once the MC has detected that the model is not running, it extracts data from the model which it passes to the expert system for a decision. The decision is returned to the simulation model via the MC. Some effort was required to ensure that this sequence of events was adhered to. A particular difficulty was encountered in detecting whether the Witness model had stopped running before seeking a decision from XpertRule. If Witness could act as an OLE client it could call XpertRule directly, removing the need for the MC. This would have simplified the linking of the packages significantly.

Having developed the interface between the two packages, the model was used in two ways.

Mode 1:

Developing decision rules directly in Xpertrule: An expert was interviewed in order to elicit information on how he would make the allocation decisions. These were then represented in XpertRule as a decision-tree as shown in Fig. 3. The allocation decision rests primarily on the capacity available in the bus. If this is less than or equal to 20 Passengers, then the bus can be allocated to any one of the lanes. An attempt is made to allocate the bus to the smaller lanes first (lanes 1 and 4). The variables Lane 1 are set to 0 if the lane is not allocated, or to the number of the bus (the first bus to arrive is numbered 1 etc.) that is currently allocated to that lane. Buses that require more than 20 Passengers can only be allocated to lanes 2 and 3. The outcome is the number of the lane to which the bus is to be allocated. If no lane is available, then the outcome is 0.

Mode 2:

Learning decision rules from examples supplied via the simulation: The simulation model was run and at a

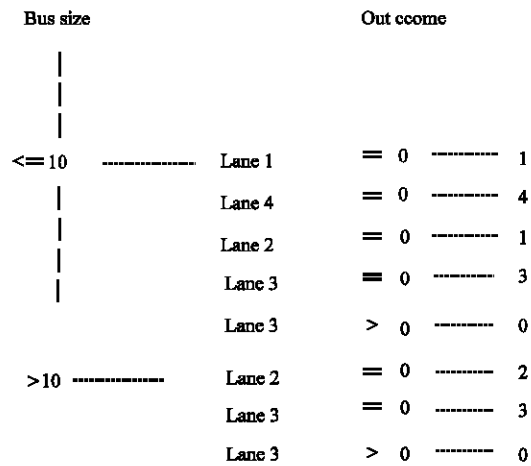


Fig. 3: Decision tree developed from knowledge elicitation

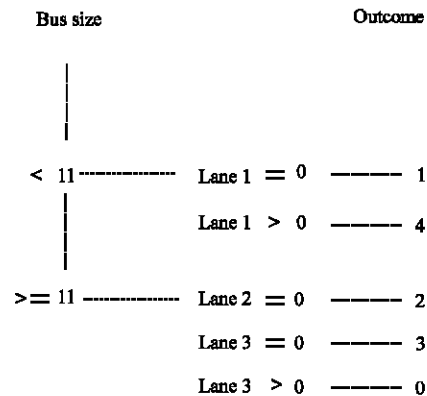


Fig. 4 Decision Tree Induced from Examples

decision point the user was prompted for an allocation decision. These decisions were logged in a data file along with five state variables: the number of items to be loaded on the bus (bus Size) and whether each of the four lanes are already allocated (Lane 1). These were then used to train the expert system. With as few as 40 examples it was possible for the expert system to obtain approximately the same decision-tree as derived in model as shown in Fig. 4 difference occurred because no instances of buses requiring 20 or less Passengers being allocated to bay 2 or 3, or indeed, not being allocated to a lane, were encountered in the examples; this was considered possible in mode 1.

CONCLUSIONS

What this simple example demonstrates is that it is technically feasible to link an expert system with a simulation model, on the same PC, to represent human decision making. The approach is particularly likely to

reap benefits when used in the second mode described above. Experts may not always be able to clearly define how they go about making complex decisions. However, by presenting them with a set of examples via a simulation and recording their decisions and some relevant state variables, it may be possible to elicit their decision making process by training an expert system. Indeed, this was possible in the loading bay example.

Having trained the expert system it could be used in three ways:

- To run the simulation model without the need for intervention from the human decision maker
- To train decision makers: either novice decision makers or potentially established decision makers by comparing the approach of different experts
- To operate the real-life facility For the first of these applications the aim of the expert

System is to represent the human decision maker as accurately as possible. This contrasts with the usual aim of an expert system, which is to make the best decisions possible, not to match the standard of the expert precisely.

REFERENCES

- Balmer, D. and R.J. Paul, 1986. CASM. The right environment for simulation. *J. Operat. Res. Soc.*, 37: 443-452.
- Deslanders, V. and H. Pierreval, 1991. An expert system prototype assisting the statistical validation of simulation models. *Simulation*, 56: 79-89.
- Doukidis, G. and R. Paul, 1985. Research into expert systems to aid simulation model formulation. *J. Operat. Res. Soc.*, 36: 319-325.
- Doukidis, G.I., 1987. An anthology on the homology of simulation with artificial intelligence. *J. Operat. Res. Soc.*, 38: 701-712.
- Doukidis, G.I. and M.C. Angelides, 1994. A framework for integrating artificial intelligence and simulation. *Artificial Intelligence Review*, 8: 55-85.
- Flitman, A.M. and R.D. Hurriion, 1987. Linking discrete-event simulation models with expert systems. *J. Operat. Res. Soc.* 38: 723-734.
- Hart, A., 1987. Role of Induction in Knowledge Elicitation. In *Knowledge Acquisition for Expert Systems: A Practical Handbook*. Kidd, A. (Ed.), New York: Plenum, pp: 165-189.
- Hurriion, R.D., 1991. Intelligent Visual Interactive Modeling. *Eur. J. Operat. Res.*, 54: 349-356.
- Hurriion, R.D., 1993. Representing and learning distributions with the aid of a neural network: An expert systems approach to simulating the human decision maker. *J. Operat. Res. Soc.*, 44: 1013-1024.
- Lyu, J. and A. Gunasekaran, 1997. An intelligent simulation model to evaluate scheduling strategies in a steel company. *Int. J. Systems Sci.*, 28: 611-616.
- Mathewson, S., 1989. Simulation Support Environments. In: *Computer Modeling for Discrete Simulation*. Pidd, M. (Ed.), Chichester, UK: Wiley, pp: 57-100.
- O'Keefe, R.M., 1989. The Role of Artificial Intelligence in Discrete-Event Simulation. In: *Artificial Intelligence Simulation and Modeling*. Widman, L.E., K.A. Loparo and N. R. Neilsen (Ed.), New York: Wiley, pp: 359-379.
- Waterman, D.A., 1986. *A Guide to Expert Systems*. Reading, Mass: Addison-Wesley.
- Williams, T., 1996. Simulating the Man-in-the-Loop. *OR Insight*, 9: 17-21.